Built-in Determined Sub-key Correlation Power Analysis

Yuichi Komano, Hideo Shimizu, Shinichi Kawamura

Toshiba Corporation,
1, Komukai Toshiba-cho, Saiwai-ku, Kawasaki 212-8582, Japan
{yuichi1.komano,hideo.shimizu,shinichi2.kawamura}@toshiba.co.jp

Abstract. Correlation power analysis (CPA) is a well-known attack against cryptographic modules with which an attacker evaluates the correlation between the power consumption and the sensitive data candidate calculated from a guessed sub-key and known data (plaintext or ciphertext). This paper enhances CPA to propose a new general power analysis, *built-in determined sub-key CPA* (BS-CPA), that finds a new sub-key by using the previously determined sub-keys recursively to compute the sensitive data candidate and to increase the signal-to-noise ratio in its analysis. BS-CPA is powerful and effective when the multiple sbox outputs (or corresponding data) are processed simultaneously as in the hardware implementation. We apply BS-CPA to the power consumption traces provided at the DPA contest and succeed in finding DES keys using less traces than the original CPA does.

Keywords: Side channel attacks, Power analysis, CPA, Hamming weight and Hamming distance models, DPA contest, DES

1 Introduction

Background: Side-channel attacks use instantaneous physical observables, such as timing [10], power consumption [11] or electromagnetic radiation [6], to reveal a secret key from a cryptographic module without leaving traces. Since Kocher et al. [11] introduced *simple power analysis* (SPA) and (higher-order) *differential power analysis* ((HO-)DPA), a great deal of research has been done to investigate attacks [12, 3, 5, 4] and countermeasures[7, 2, 13, 9, 14].

Correlation power analysis (CPA, [3]) is an enhancement of first-order DPA (DPA, for short). It finds a part of round key (sub-key) by calculating the correlation between the power consumption and the sensitive data candidate, which is a function of a guessed sub-key and known data such as plaintext and ciphertext. CPA and DPA are not only of theoretical interests but real threats to the cryptographic module. In fact, in the

DPA contest [1], sophisticated attacks of these kinds have been reported. The DPA contest is a competition for the analysis. It provides a number of power consumption traces (more than eighty thousands traces) and calls for analysis requiring fewer traces for successful attack. Recently, several results have been reported to find DES key from 310 traces (by enhanced CPA), 232 traces (by enhanced CPA with 12-bit key search) and 135 traces (by CPA with a very special file order) until March 16th, 2009.

In the DPA contest, since the abundant power consumption traces are provided in advance, we may find a suitable preselection of traces for analysis (a special file order, etc.) or a specific analysis for the preset.

Motivation and Our Contribution: This paper proposes a *general* and *efficient* analysis which finds a key using not so many traces in the non-specific file order. This is motivated by the fact that the fair estimation with such analysis helps us to design secure cryptographic modules rapidly. Note that the specific preset of traces is applicable to our analysis and reduces the number of required traces furthermore.

Our proposal, named *built-in determined sub-key correlation power analysis* (BS-CPA), uses such restricted number of traces recursively. BS-CPA is an extension of CPA which finds a new sub-key by using the previously determined sub-keys for computing the sensitive data candidate to increase the signal-to-noise ratio (SNR). BS-CPA is powerful and effective when the device processes multiple sbox outputs (or corresponding data such as left outputs in DES) simultaneously as is the case for a hardware implementation. We apply BS-CPA to the power consumption traces provided at the DPA contest and succeed in finding DES key with 164 traces[1]. As of March 16th, 2009, our analysis is only one that determines the whole key with less than 200 traces in *non-special file order* (we refer the database file order in the DPA contest).

The rest of this paper is organized as follows. At first, we prepare some notations we use in section 2, and then, we review CPA in section 3. Section 4 explains our proposal, *BS-CPA*, and discusses its aspects. Section 5 applies BS-CPA to hardware DES implementation to demonstrate the power of BS-CPA with some experimental results. Finally, section 6 concludes this paper.

[1] The DPA contest regards an attack as successful if the guessed key is unchanged while more 100 traces are processed.

Related Work: Hanley et al. [8] discusses CPA with large word size. Their aim is to decrease the number of CPA traces while our aim is to decrease the number of power traces required for CPA.

Let us consider CPA to hardware DES implementation. The largest correlation can be obtained if 2^{48} CPA traces are generated with guess of 2^{48} round keys; however, it is computationally infeasible. They achieve the largest correlation with only 2^{11} CPA traces as follows. We first guess the first 6-bits sub-key (for S_1) and generate 2^6 CPA traces to determine four candidates with the largest correlation. After that, we include these four candidates in guessing first two sub-keys (for S_1 and S_2). This requires $2^6 \times 4$ CPA traces not 2^{12} CPA traces normally required for guessing first two sub-keys. This process is continued until the whole eight sub-keys are determined. It requires $2^6 + 7(2^6 \times 4) = 2^{11}$ CPA traces.

The idea to use previously determined sub-keys in finding another sub-key is similar to ours; however, our aim is different to theirs. Their aim is to find the large correlation with fewer CPA traces. Their method may require many power traces in finding the first sub-key (since their method fixed the target previously) and does not mention whether power traces are reused in finding another sub-key. BS-CPA searches a sub-key determined by fewer power traces in parallel and reuses the power traces in finding another sub-key to strategically decrease the number of power traces required for CPA.

2 Notations

Let $P_i = (p_i(t_1), p_i(t_2), \cdots, p_i(t_m))$ be power consumption traces including m points where i indicates the order of trace. We denote the maximum number of trace by n_{tr}.

This paper deals with correlation power analyses to symmetric key cryptosystem that has non-linear transformation called *sbox*. We denote the number of sboxes used in one round of the cryptosystem by n_{sb}. For instance, $n_{sb} = 8$ for DES and $n_{sb} = 16$ for AES. We also represent sb-th sbox as $sbox_{sb}$.

CPA guesses a part of round key (sub-key) $key^{(sb)}$ corresponding to $sbox_{sb}$ and computes the ℓ-bit sensitive data candidate $b_i^{(sb)}$ from guessed $key^{(sb)}$ and known input X_i for i-th trace. We denote the number of key candidate for one sbox by n_{key}. For example, $n_{key} = 64$ for DES and $n_{key} = 256$ for AES.

CPA then calculates the correlation and finds the sub-key if the correlation exceeds or equals some threshold $th^{(sb)}$. Note that, the threshold may be maximum $max^{(sb)}$ of correlation over all key_{sb} and t_j, or three times of the standard deviation of correlation over key_{sb} at t_j. Throughout this paper, without loss of generality, we regard $max^{(sb)}$ as $th^{(sb)}$.

3 CPA

To determine $key^{(sb)}$, the original CPA utilizes the correlation between Hamming weight of $b_i^{(sb)}$ and $p_i(t_j)$ for some j. We give a sketch of CPA below.

1. for sb from 1 to n_{sb}
2. for $key^{(sb)}$ from 0 to $n_{key} - 1$
3. compute the correlation $\mathsf{cpa}(key^{(sb)}, t_j)$ below
4. if $\mathsf{cpa}(key^{(sb)}, t_j) = max^{(sb)}$,
5. then store $key^{(sb)}$ and move to next sb (go to line 1)
6. end for
7. end for
8. output $key^{(1)}, \cdots, key^{(n_{sb})}$

Here, the correlation $\mathsf{cpa}(key^{(sb)}, t_j)$ is evaluated by

$$\mathsf{cpa}(key^{(sb)}, t_j) = \frac{1}{n\sigma_{h^{(sb)}}\sigma_{p(t_j)}} \sum_{i=1}^{n} (h_i^{(sb)} - \overline{h^{(sb)}})(p_i(t_j) - \overline{p(t_j)})$$

where $h_i^{(sb)} \in [0, \ell]$ is the Hamming weight of sensitive data candidate $b_i^{(sb)} \in \{0,1\}^{\ell}$. $\overline{x}$ and σ_x mean the average and the standard deviation of random variable x, respectively:

$$\overline{h^{(sb)}} = \frac{1}{n}\sum_{i=1}^{n} h_i^{(sb)}, \quad \sigma_h^{(sb)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(h_i^{(sb)} - \overline{h^{(sb)}}\right)^2}$$

$$\overline{p(t_j)} = \frac{1}{n}\sum_{i=1}^{n} p_i(t_j), \quad \sigma_{p(t_j)} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(p_i(t_j) - \overline{p(t_j)}\right)^2}$$

CPA is known to be stronger than DPA. In the above procedure, loops for sb and $key^{(sb)}$ can be proceeded in parallel if $\mathsf{cpa}(key^{(sb)}, t_j)$ is stored in an array $\mathsf{cpa}[n_{sb}][n_{key}]$.

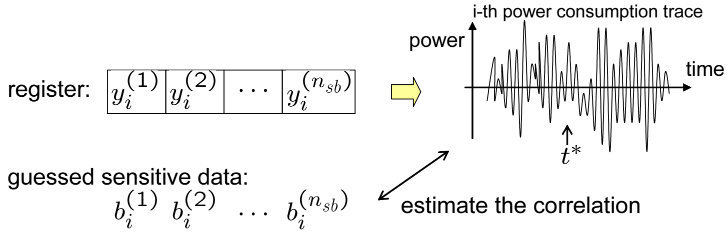

Fig. 1. Relation among sensitive data, its candidate, and power trace

4 Built-in determined Sub-key CPA: BS-CPA

As described in the previous section, CPA determines $key^{(sb)}$ for each $sbox_{sb}$ independently. *BS-CPA* diverts the previous results ($key^{(1)}, \cdots, key^{(sb-1)}$, for example) to increase the signal-to-noise ratio (SNR) in findind the next sub-key ($key^{(sb)}$). This method is particularly effective for a hardware implementation where multiple sbox outputs (and corresponding data such as the left outputs of DES) are processed simultaneously.

4.1 Intuition

We first give the intuition of our analysis. Traditional differential and correlation power analyses, such as DPA and CPA, follow a simple strategy allowing us to find $key^{(sb)}$ for each $sbox_{sb}$ in parallel. It seems to come from the facility of programming (straightfoward algorithm) and the admissibility in accessing a number of traces.

Our aim is to give the analysis that finds the sub-key under the limited circumstance such that an attacker is fed with not so many traces or disallowed to choose input (see section 1). Therefore, we make the sacrifice of facility of programming to decrease the number of required traces. Let us consider the SNR with Figure 1.

The power consumption relates to the Hamming weight (in many CPUs) and/or Hamming distance (in CMOS logic) of the sensitive data. Here, the Hamming distance of sensitive data means the Hamming weight for XOR of sensitive data at the two consecutive clock cycles. For simplicity, since the experiments (section 5) use the Hamming distance model in guessing the sensitive data, we discuss only the Hamming distance model.

Assume that register holds sensitive data $y_i^{(1)}, \cdots, y_i^{(n_{sb})}$ (for i-th trace) and their Hamming distances induces the power consumption at time t^*. In CPA, an attacker guesses the sensitive data as $b_i^{(sb)}$ and computes the correlation between $b_i^{(sb)}$ and $p(t^*)$.

If the correlation is taken over both Hamming distance of $b_i^{(sb)}$ and $p_i(t^*)$ for some sb, the portion of power consumption induced by other sensitive data $y_i^{(1)}, \cdots, y_i^{(sb-1)}, y_i^{(sb+1)}, \cdots, y_i^{(n_{sb})}$ behave as a noise and the SNR of analysis decreases. The number of required traces for determining the whole key is the largest one for determining $key^{(sb)}$ for $sbox_{sb}$ that leaks less information among the sboxes.

The multiple guess of key_{sb} and $key_{sb'}$ increases the SNR. If the attacker guesses two sensitive data $b_i^{(sb)}$ and $b_i^{(sb)'}$ and calculates the correlation between sum of the Hamming distance of two sensitive data candidates and $p_i(t^*)$, the portion of power consumption induced by $y_i^{(sb')}$ change from a noise to a signal. Hence, fewer traces are required for determining the key compared to single guess of $b_i^{(sb)}$ as above.

The more multiple guess the attacker makes, the fewer traces are sufficient to determine the key; however, multiple guess leads to the increase of sub-key candidates. Our strategy is to feedback the information of determined sub-key recursively to make multiple guess of several sensitive data with only one $key^{(sb)}$.

4.2 Procedure

This subsection gives a description of BS-CPA. The numbers of required traces for determining $key^{(sb)}$ differ in each sb depending on leakage of corresponding sensitive data b^{sb}. Since an attacker is unable to find which $key^{(sb)}$ can be determined with the least number of traces, we prepare an array $\mathsf{bucpa}[n_{sb}][n_{key}]$ to store the $\mathsf{bs\text{-}cpa}(keys^{(SB)}, keys^{(sb)}, t_j)$ below to search $key^{(sb)}$ in parallel. We use two lists $I_0 = \{sb\}$ and $I_1 = \{(SB, key^{(sb)})\}$ including indexes of sboxes corresponding to undetermined sub-keys and pairs of index and determined sub-key, respectively.

1. set $I_0 = \{1, \cdots, n_{sb}\}$ and $I_1 = \phi$
2. if I_0 is empty, then output I_1
3. else for all $sb \in I_0$
4.   for all sub-key candidate $key^{(sb)}$
5.   compute the correlation $\mathsf{bs\text{-}cpa}(keys^{(SB)}, key^{(sb)}, t_j)$
6.   if $\mathsf{bs\text{-}cpa}(keys^{(SB)}, key^{(sb)}, t_j) = max^{(sb)}$,
7.   then remove sb from I_0, add $(sb, key^{(sb)})$ to I_1, and go to 2.
8.   end for
9. end for

Here, the correlation $\mathsf{bs\text{-}cpa}(keys^{(SB)}, key^{(sb)}, t_j)$ is evaluated by the following equation. Let $N \in [0, n_{sb} - 1]$ be the number of elements in I_1.

$$\begin{aligned}
&\mathsf{bs\text{-}cpa}(keys^{(SB)}, key^{(sb)}, t_j) \\
&\quad = \frac{1}{n\sigma_{H^{(SB,sb)}}\sigma_{p(t_j)}} \sum_{i=1}^{n} (H_i^{(SB,sb)} - \overline{H^{(SB,sb)}})(p_i(t_j) - \overline{p(t_j)})
\end{aligned}$$

where $H_i^{(SB,sb)} \in [0, \ell \times (N+1)]$ is the sum of Hamming distances of sensitive data candidates $b_i^{(SB_1)}, \cdots, b_i^{(SB_N)}, b_i^{(sb)} \in \{0,1\}^\ell$ derived from the input X_i (for the i-th trace), the previously determined sub-keys $key^{(SB_1)}, \cdots, key^{(SB_N)}$ in I_1 and the sub-key candidate $key^{(sb)}$, respectively. $p_i(t_j)$ and $\overline{p(t_j)}$ are the same as those described in section 3.

4.3 Variations

The idea of built-in determined sub-keys that increases the SNR can be applied to other power analyses such as *built-in determined sub-key DPA* (BS-DPA), *built-in determined sub-key higher-order DPA* (*BS-HO-DPA*), *built-in validate sub-key zero-offset second-order DPA* (*built-in validate sub-key ZO-2DPA*) (BS-ZO-2DPA), etc. Especially, since the SNR is critical in HO-DPA (that requires a lot of traces for analysis), our idea leads to practical analyses.

The combination of built-in validate sub-key and multiple sub-key guess (starting from guessing both $key_i^{(sb)}$ and $key_i^{(sb)'}$; for instance, 12-bits sub-keys for DES) is also one of possible solutions to decrease the number of required traces, however, more memory space is required.

5 Experimental Results

This section applies BS-CPA to the hardware DES implementation. We use the power consumption traces provided at the DPA contest. We first discuss a strategy for attacking the hardware DES implementation with BS-CPA.

5.1 BS-CPA against Hardware DES Implementation

With the hardware, DES is usually implemented by a loop architecture having two 32-bits registers holding left and right outputs, respectively. Figure 2 illustrates the data flow of DES after the 15th round. In the loop

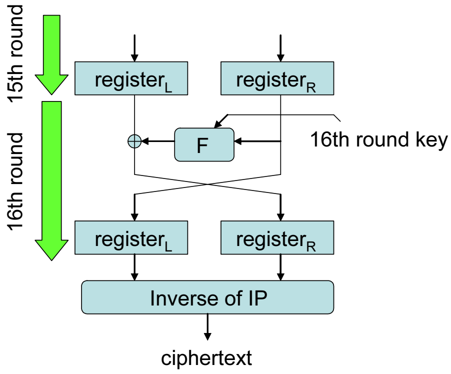

Fig. 2. Data flow of DES

architecture, the left registers $\mathsf{register}_L$ (resp. right registers $\mathsf{register}_R$) described at the end of each round are identical. F is the round function consisting of 4-bits to 6-bits extension permutation, eight 6-bits key addition, eight 6-bits to 4-bits sboxes and 32-bits permutation.

Let us consider the i-th trace. At the clock end of the 15th round, $\mathsf{register}_L$ holds the left output of the 15th round ($L_{15,i}$) which equals the XOR of right output of the 16th round ($R_{16,i}$) and output of the round function $F(key^{(sb)}, L_{16,i})$. On the other hand, at the end of the 16th round, $\mathsf{register}_L$ holds the left output of the round ($L_{16,i}$). We regard the distance of left outputs of the 15th and 16th rounds as the sensitive data, namely,

$$b_i^{(sb)} = L_{15,i}^{(sb)} \oplus L_{16,i}^{(sb)} = R_{16,i}^{(sb)} \oplus F^{(sb)}(key^{(sb)}, L_{16,i}) \oplus L_{16,i}^{(sb)}.$$

Here, $L_{15,i}^{(sb)}$, etc., represent bits corresponding to $sbox_{sb} \in \{sbox_1, \cdots, sbox_8\}$ through the permutation of the round function F.

In this analysis, we store the correlation in array, consisting of $n_{sb} = 8$ times $n_{key} = 64$ elements, and calculate multiple correlation for each sbox simultaneously. If some sub-key is determined, we roll-back the trace to the first one, use the determined sub-key for re-computing the sensitive data candidates (its length is enlarged by ℓ) and try to find another sub-key.

Note: It seems to be able to increase the SNR by adding the Hamming distance of data for the right register $\mathsf{register}_R$. $\mathsf{register}_R$ holds $R_{15,i} = L_{16,i}$ and $R_{16,i}$ at the clock end of the 15th and 16th clock cycles, respectively. We regard $\hat{b}_i^{(sb)} = L_{16,i} \oplus R_{16,i}$ as another sensitive data and replace $H_i^{SB,sb}$ in $\mathsf{cpa}(key^{(SB)}, key^{(sb)}, t_j)$ with $\widehat{H}_i^{SB,sb}$ that is the sum of

Hamming weights of $b_i^{(sb)}$ and $\hat{b}_i^{(sb)}$. Note that since $b_i^{(sb)}$ and $\hat{b}_i^{(sb)}$ are computed by XOR and $\widehat{H}_i^{SB,sb}$ is their Hamming distance. We tried this idea to our experiments; however, the result was not improved (one or two more traces are required compared to the analysis using only $\mathsf{register}_L$ for computing the sensitive data candidates).

5.2 Experiments

The DPA contest provides the power consumption traces of hardware DES implementations (secmatv1_2006_04_0809, secmatv3_20070924_des, and secmatv3_20071219_des).

Our analysis is performed under the following condition.

- We apply BS-CPA.
- We guess the 6-bits sub-key for $sbox_{sb}$, not 12-bits sub-keys block for $sbox_{sb}$ and $sbox_{sb'}$.
- We regard the Hamming distance of $\mathsf{register}_L$ as the sensitive data.
- We use the traces of secmatv1_2006_04_0809.
- We follow the database file order, *not the special file (suitable) order.*
- We evaluate whole points, *not the narrowed points.*
- We compress 15 points to one point.

Table 1. Numbers of required traces for BS-CPA and CPA

order	sbox	bs-cpa
1	$sbox_2$	65
2	$sbox_3$	30
3	$sbox_4$	46
4	$sbox_7$	44
5	$sbox_8$	49
6	$sbox_6$	64
7	$sbox_5$	59
8	$sbox_1$	48
max	—	65

order	sbox	cpa
1	$sbox_2$	65
2	$sbox_4$	78
3	$sbox_1$	90
4	$sbox_8$	91
5	$sbox_3$	153
6	$sbox_5$	236
7	$sbox_7$	268
8	$sbox_6$	280
max	—	280

Table 1 shows the result of BS-CPA and traditional CPA. Both analyses first find $key^{(2)}$ for $sbox_2$ with 65 traces. After that, these analyses take different ways.

BS-CPA roll-backs (rewinds) the traces and uses $key^{(2)}$ to find another sub-key; and it secondly finds $key^{(3)}$ with 30 traces. As shown in table

1, it recursively finds $key^{(4)}, key^{(7)}, \cdots, key^{(1)}$ with 46, 44, $\cdots$, 48 traces, respectively. Since the maximum is 65, BS-CPA specifies the 56-bits key of the 16th round with 65 traces (164 traces under the policy of DPA contest).

On the other hand, CPA *does not* use the results previously obtained in the analysis in finding another sub-key. It secondly finds $key^{(4)}$ with more 13 traces (at the 75th trace), and it sequentially finds $key^{(1)}, key^{(8)}, \cdots, key^{(6)}$ with 90, 91, $\cdots$, 280 traces, respectively. Since the maximum is 280, CPA determines the 56-bits key of the 16th round with 280 traces (379 traces under the policy of DPA contest).

Therefore, we can conclude BS-CPA can determine the whole key with much less power consumption traces than CPA requires.

Note: If we follow some results of DPA contest to narrow the point from 14450 to 14550, then the number of required traces in BS-CPA decreases 65 to 44 (143 traces under the policy of DPA contest). Moreover, using a very special file order (with points from 14400 to 14500) decreases the number of required traces to 20 (119 traces under the policy); however, in order to obtain this result, BS-CPA works not greedily but strategically.

6 Conclusion

A new general power analysis, *built-in determined sub-key CPA* (BS-CPA), that uses restricted traces thoroughly was proposed. It recursively uses the previous results to increase the SNR and finds another sub-key without increasing the guess of sub-key candidates. BS-CPA is powerful and effective when the multiple sbox outputs or multiple corresponding data are processed simultaneously time as in the case for the hardware implementation. We apply this analysis to data provided at the DPA contest and succeed in finding DES key less than the original CPA does.

Acknowledgements. We would like to thank Sylvain Guilley for useful comments on previous version of this paper.

References

1. DPA Contest 2008/2009. `http://www.dpacontest.org/`.
2. Mehdi-Laurent Akkar and Christophe Giraud. An Implementation of DES and AES, Secure against Some Attacks. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 309–318, Berlin, Heidelberg, New York, 2001. Springer.

3. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29, Berlin, Heidelberg, New York, 2004. Springer.
4. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28, Berlin, Heidelberg, New York, 2003. Springer.
5. Pierre-Alain Fouque, Frederic Muller, Guillaume Poupard, and Frederic Valette. Defeating countermeasures based on randomized bsd representations. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 312–327, Berlin, Heidelberg, New York, 2004. Springer.
6. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261, Berlin, Heidelberg, New York, 2001. Springer.
7. L. Goubin and J. Patarin. DES and Differential Power Analysis (The "Duplication" Method). In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems – CHES'99*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172, Berlin, Heidelberg, New York, 1999. Springer-Verlag.
8. Neil Hanley, Robert McEvoy, Michael Tunstall, Claire Whelan, Colin Murphy, and William P. Marnane. Correlation Power Analysis of Large Word Sizes. In *ISSC (Irish Signals and System Conference)*. IET, 13-14 Sept 2007. Edinburgh, Scotland, UK.
9. Kouichi Itoh, Masahiko Takenaka, and Naoya Torii. DPA countermeasure based on the "masking method. In Kwangjo Kim, editor, *Information Security and Cryptology - ICISC 2001*, volume 2288 of *Lecture Notes in Computer Science*, pages 440–456. Springer-Verlag, Berlin, 2002.
10. Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113, Berlin, Heidelberg, New York, 1996. Springer-Verlag.
11. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In M. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397, Berlin, Heidelberg, New York, 1999. Springer-Verlag.
12. Thomas S. Messerges. Using second-order power analysis to attack DPA resistant software. In Çetin Kaya Koç and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2000*, volume 1965 of *Lecture Notes in Computer Science*, pages 238–251, Berlin, Heidelberg, New York, 2000. Springer.
13. Thomas S. Messerges. Securing the AES finalists against power analysis attacks. In Bruce Schneier, editor, *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 150–164. Springer-Verlag, Berlin, 2001.
14. Kai Schramm and Christof Paar. Higher order masking of the aes. In D. Pointcheval, editor, *Topics in Cryptology - CT-RSA 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 208–225, Berlin, Heidelberg, New York, 2006. Springer.